

Regular Expressions for Irregular Rhythms

Houssam Abbas
University of Pennsylvania
Philadelphia PA 19104, USA
habbas@seas.upenn.edu

Alena Rodionova
TU Wien
Vienna, Austria
alena.rodionova@tuwien.ac.at

Ezio Bartocci
TU Wien
Vienna, Austria
ezio@cps.tuwien.ac.at

Scott A. Smolka
Stony Brook University
Stony Brook, NY, USA
sas@cs.stonybrook.edu

Radu Grosu
TU Wien
Vienna, Austria
radu.grosu@tuwien.ac.at

ABSTRACT

Motivated by the desire to verify the correctness of algorithms for arrhythmia discrimination used in cardiac medical devices, we present a general wavelet-based characterization of *peaks* (local maxima and minima) that occur in cardiac electrograms, along with two peak-detection algorithms based on this characterization. *Peak detection* (PD) is a common signal-processing task, as peaks indicate *events of interest*, such as heartbeats (in the cardiac setting). The performance of PD thereby directly influences the correctness of the algorithms that depend on its output. We show that our wavelet-based PD algorithms (*peakWPM* and *peakWPB*) and a commercial PD algorithm from Medtronic Inc. (*peakMDT*) are easily expressible in Quantitative Regular Expressions (QREs), a formal language based on regular expressions for specifying complex numerical queries over data streams. We then study the accuracy and sensitivity of the resulting QRE-based PD algorithms on real patient data, and show that the wavelet-based *peakWPM* algorithm outperforms the other two PD algorithms, yielding results that are on par with those provided by a cardiologist.

Keywords

Peak Detection; Electrocardiograms; ICDs; Continuous Wavelet Transform; Quantitative Regular Expressions

1. INTRODUCTION

Medical devices seamlessly blend signal processing (SP) algorithms with decision algorithms such that the performance and correctness of the latter critically depends on that of the former. As such, analyzing a device’s decision making in isolation of SP offers at best an incomplete picture of the device’s overall behavior. For example, an Implantable Cardioverter Defibrillator (ICD) will first perform *Peak Detection* (PD) on its input signal, also known as an *electrogram* (see, e.g., Fig. 4). The output of PD is a timed boolean signal where a 1 indicates a peak, i.e., a local maximum or minimum, which is used by the downstream *discrim-*

inators to differentiate between fatal and non-fatal rhythms.

The detected electrogram peaks indicate when a heart-beat occurs, and the accuracy of PD directly affects the correctness of the discriminators’ decisions. Over-sensing (too many false peaks detected) and under-sensing (too many true peaks missed) can be responsible for as much as 10% of an ICD’s erroneous decisions [24], as they lead to inaccuracies in estimating the heart rate.

Motivated by the desire to verify ICD algorithms for cardiac arrhythmia discrimination, we seek a unified formalism for expressing and analysing the SP and discrimination tasks commonly found in ICD algorithms. This paper focuses on peak detection because of the important role it plays in arrhythmia discrimination, and because PD is a fundamental SP primitive in its own right.

The signals we analyze (electrograms) have *time-varying frequency content*, thereby motivating us to consider the PD problem in the *wavelet domain* [19]. We therefore present general, wavelet-based characterizations of peaks in time-series data (i.e., signals), with and without a *blanking period*: a period of time, typically defined by a cardiologist, during which at most one peak can occur.

An implementation of our wavelet-based characterizations of peaks would require one to store different values of the input (wavelet-domain) signal, and to perform complex numerical operations on the signal. It is therefore unlikely that these peak characterizations can be expressed succinctly (if at all) in temporal logic (TL) [21], despite the increasingly sophisticated variety of TLs that have appeared in the literature [17, 10, 5]. This is the case even if we use a quantitative semantics [11]. We explore this matter further in Section 1.1.

We therefore propose the use of Quantitative Regular Expressions (QREs) to describe wavelet-based peak detection. QREs are a formal language based on classical regular expressions for specifying complex numerical queries on data streams [2]. QREs’ ability to interleave user-defined computation at any nesting level of the underlying regular expression, and the fact that their design is parameterized by the domain of their input data (time, frequency or other domains), gives them significant expressive power. We show that our wavelet-based peak detection algorithms are easily expressed in QREs (Section 5). We also formalize a commercial peak detector as a QRE. This allows us to readily study the accuracy and sensitivity of the resulting algorithms on real patient electrograms.

As ongoing work, we have used QREs to capture a number of ICD discriminators. This makes us highly confident that QREs will serve as the unifying formalism we seek for expressing and analyzing the SP and discrimination tasks

found in ICD algorithms.

In summary, our main contributions are the following:

- We present general wavelet-based characterization of peaks along with two PD algorithms (*peakWPM* and *peakWPB*) based on this characterization.
- We show that our wavelet-based PD algorithms, and a commercial PD algorithm found in defibrillators currently on the market (*peakMDT*), are easily expressible in Quantitative Regular Expressions (QREs).
- We study the accuracy and sensitivity of the resulting QRE-based PD algorithms on real patient data and show that the wavelet-based algorithm *peakWPM* outperforms the other two PD algorithms, yielding results that are on par with those provided by a cardiologist.

The rest of the paper is organized as follows. Section 2 serves as an introduction to signal processing in the wavelet domain. Section 3 presents *WPM* and *WPB*, our wavelet-based characterizations of peaks without and with a blanking period. Section 4 offers a primer on QREs, while Section 5 presents *peakWPM* and *peakWPB*, our QRE-based formalization of *WPM* and *WPB*, respectively. Section 6 compares the performance of the *peakWPM*, *peakWPB* and *peakMDT* PD algorithms on real patient data. Section 7 discusses related work. Section 8 offers our concluding remarks and a number of directions for future work.

1.1 Different Logics for Different Tasks

We will briefly illustrate how different temporal logics are needed for different ICD tasks, thus prompting the use of QREs. The details of the syntax in the following formulas is not important, we offer them merely for illustration. For example, to express that the number of heartbeats in a one-minute interval must be between 90 and 120 requires the use of a counting modality, such as is used in Counting MTL (p denotes a heartbeat):

$$\Box \left(C_{[0,59]}^{\geq 90} p \wedge C_{[0,59]}^{\leq 120} p \right)$$

This number of heartbeats can be very simply computed using a QRE: simply match $60f$ samples (where f is the sampling rate), and output the total number of beats as the final cost:

$$iter_{60f} - add(p?1)$$

To express that the heart rate increases by at least 20% when measured over consecutive and disjoint windows of 4 beats requires explicit clocks, such as those used in XCTL, since the beat-to-beat delay is variable (the x_i 's are clocks):

$$p \wedge (x_1 = T) \wedge \Diamond(p \wedge \dots \Diamond(p \wedge (x_9 = T) \wedge (x_5 - x_1) * 0.8 \geq x_9 - x_5))$$

This can be expressed in QREs as:

$$\text{suddenOnset} := \text{split-inc}(\text{fourBeats}, \text{fourBeats})$$

where *fourBeats* matches four consecutive beats, and *inc*(x, y) is an operator that returns True whenever $0.8x \geq y$.

As a final example, expressing that a threshold $a > 0$ for peak detection is reset as a function of the previous peak value requires us to remember the value of that peak. This necessitates memory registers or *freeze quantification*

of state variables, as used in Constraint LTL with Freeze Quantification CLTL[↓] (\downarrow denotes the freeze quantifier):

$$\Box(y > a \implies \downarrow_{BL=1} \Diamond(\varphi_{\text{local-max}} \implies a = 0.8z_2))$$

$$\varphi_{\text{local-max}} := \downarrow_{z_1=y} X(\downarrow_{z_2=y} X(z_2 > z_1 \wedge z_2 > y))$$

This variety of logics required for these tasks, all of which are fundamental building blocks of ICD operation, means that a TL-based approach to the problem is unlikely to yield a unifying view, whereas QREs clearly do.

2. WAVELET REPRESENTATIONS

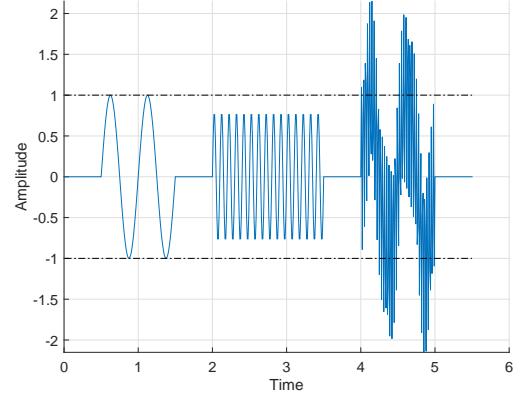


Figure 1: Signal in time domain

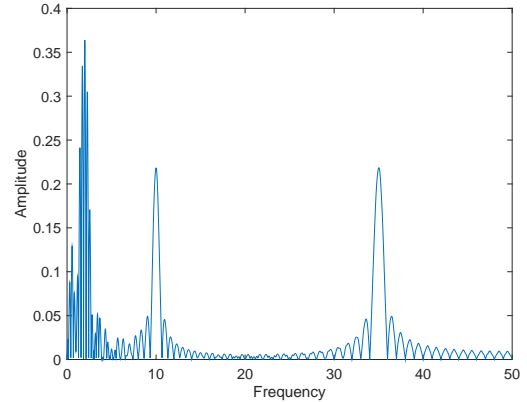


Figure 2: Signal of Fig. 1 in Fourier domain

To analyze the properties of a complex stationary signal $x : \mathbb{R}_+ \rightarrow \mathbb{R}$, it is convenient to decompose it into its harmonics, which are sinusoids with a particular phase and frequency. Since $e^{iwt} = \cos wt + i \sin wt$, the signal can equivalently be decomposed into a sum of complex exponentials. This is accomplished through the use of the *Fourier Transform* [19]:

$$\hat{x}(w) = \int_{-\infty}^{+\infty} x(t) e^{-iwt} dt, \quad x(t) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} \hat{x}(w) e^{iwt} dw \quad (1)$$

$\hat{x}(w)$ is called the Fourier transform of $x(t)$. Eq. (1) proves that the transform is invertible and so the signal is uniquely determined by either its time-domain or Fourier-domain description. Parameter w is the frequency: $|\hat{x}(w)|$ gives the power of the signal at frequency w . Thus, the larger $|\hat{x}(w)|$, the more energy is contained at that frequency.

For example, the signal $x(t)$ in Fig. 1 contains three distinct regimes of variation at different frequencies. The first regime contains two periods of a slow (5Hz frequency) sinusoid, the second contains fifteen periods of fast (10 Hz) sinusoid, and the third contains the superposition (linear composition) of the first with a very fast (35 Hz) sinusoid. The fast harmonics in the last regime can be thought of as constituting noise. The Fourier transform of $x(t)$ is depicted in Fig. 2 and shows three large components at the frequencies 5, 10 and 35 Hz. The frequency domain provides a very intuitive way of capturing the harmonics of a signal, and the Fourier Transform is a tool for bringing these harmonics to light. In contrast, writing a temporal formula capturing a signal of one or several frequencies is impractical.

The Fourier transform gives the frequency content, but *does not localize it in time*. Looking at Fig. 2, one cannot further specify whether the 35 Hz component occurs early or late in the signal. A solution would be to take a Fourier transform in a sliding window of the signal, but this raises questions of which windowing function to use, how wide it should be, the amount of overlap between windows, etc.

The *wavelet transform* [19] answers these questions in a rigorous manner, and is one of the most common and powerful means for *time-frequency analysis* of signals, with applications in almost all areas of engineering. In many CPS (cyber-physical systems) applications, such as in geodesy, electrical and mechanical engineering, the properties of the signals under observation, such as the EGM signals in this paper, are both time- and frequency-dependent. As such, we seek formal notations that can capture complex specifications or queries on wavelet-domain functions, and perform tasks that cannot be performed by time- or frequency-domain methods alone.

The idea in wavelet analysis is to have a sliding window in which to perform the signal decomposition. The size of the window is inversely proportional to the (instantaneous) frequency of the signal. A rectangular window introduces artifacts, while a Gaussian-like window does not, due to its smoothness properties. A wavelet transform W_x of signal x uses a *family* of analysis functions $\{\Psi_s\}_{s>0}$ called *wavelets*:

$$W_x(s, t) = \int_{-\infty}^{+\infty} x(\tau) \Psi_s(\tau - t) d\tau \quad (2)$$

This family is obtained by scaling (by $1/\sqrt{s}$) and dilating (by $1/s$) a *mother wavelet* $\psi(t)$: $\Psi_s(t) = \frac{1}{\sqrt{s}} \psi(\frac{t}{s})$. Different choices of the mother wavelet have different benefits. A common choice with nice properties for peak detection is the n^{th} derivative of a Gaussian, that is: $\psi(t) = \frac{d^n}{dt^n} G_{\mu, \sigma}(t)$, where $G_{\mu, \sigma}$ is a Gaussian function with mean μ and variance σ^2 , and n is a user-chosen parameter. Eq. (2) is known as a *Continuous Wavelet Transform* (CWT), and $W_x(s, t)$ is known as the *wavelet coefficient*.

Parameter s in the wavelet ψ_s is known as the *scale* of the analysis. It can be thought of as the analogue of frequency for Fourier analysis. A smaller value of s (in partic-

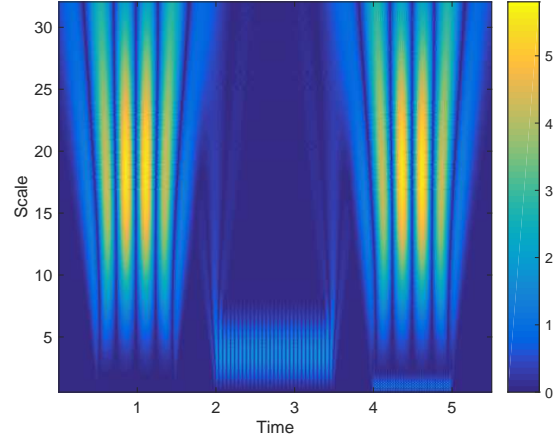


Figure 3: CWT spectrogram of signal in Fig. 1

ular $s < 1$) *compresses* the mother wavelet as can be seen from the definition of Ψ_s , so that only values close to $x(t)$ influence the value of $W_x(s, t)$ (see Eq. (2)). Thus, at smaller scales, the wavelet coefficient $W_x(s, t)$ captures *local* variations of x around t , and these can be thought of as being the higher-frequency variations, i.e., variations that occur over a small amount of time. At larger scales (in particular $s > 1$), the mother wavelet is *dilated*, so that $W_x(s, t)$ is affected by values of x far from t as well. Thus, at larger scales, the wavelet coefficient captures more *global* variations of x , such as variations over large periods of time.

By performing the wavelet analysis over many scales, we obtain a rich understanding of variations within the signal, the scales at which they occur, and the energy content of such variations. Fig. 3 shows $|W_x(s, t)|$ for the signal of Fig. 1. Such a plot is known as a *spectrogram*. Time t runs along the x-axis and scale s runs along the y-axis. (The spectrogram is computed numerically so it is effectively a discrete structure). Brighter colors indicate larger values of coefficient magnitudes $|W_x(s, t)|$. It is easier to see now that early in the signal, mid- to low-frequency content is present (bright colors mid- to top of spectrogram), followed by higher-frequency variation (brighter colors at smaller scales), and near the end of the signal, two frequencies are present: mid-range frequencies (the bright colors near the middle of the spectrogram), and very fast, low amplitude oscillations (the light blue near the bottom-right of the spectrogram). The projection of the spectrogram on the y-axis displays the frequency content of the signal (e.g., the Fourier transform) and the projection of the spectrogram on the x-axis displays the time content of the signal.

3. PEAKS IN THE WAVELET DOMAIN

The wavelet characterization of peaks uses classical work in singularity detection [18], and relaxes it to allow some numerical errors in the implementation. Consider the signal in Fig. 4 and its CWT *spectrogram* $|W_x(s, t)|$ (Fig. 5). Recall that $|W_x(s, t)|$ is a measure of signal power at (s, t) . At larger scales, one obtains an analysis of the low-frequency variations of the signal, which are unlikely to be peaks. Indeed, peaks are characterized by a rapid change in signal value, which is a higher-frequency dynamic. At smaller

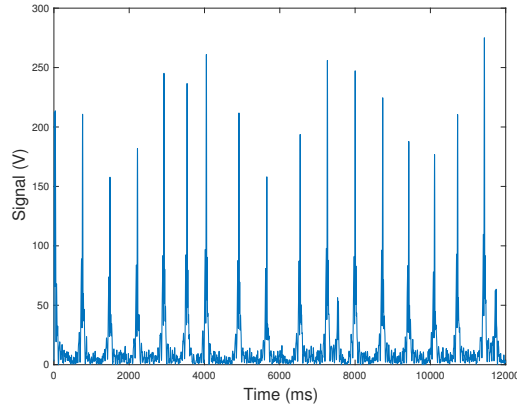


Figure 4: Rectified EGM during normal rhythm

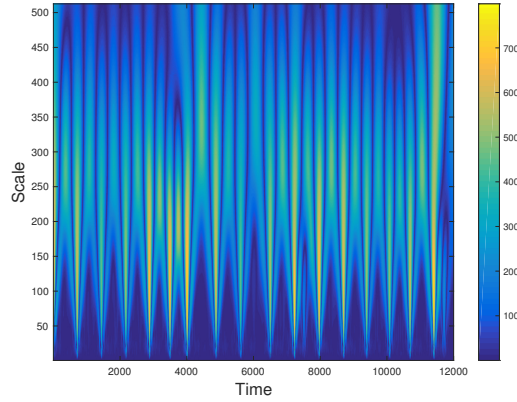


Figure 5: CWT spectrogram of EGM in Fig. 4

scales, one obtains an analysis of high-frequency components of the signal, which will include both peaks and noise. These remarks can be put on solid mathematical footing [19, Ch. 6]. **Therefore, for peak detection we must start by querying CWT coefficients that occur at an appropriately chosen \bar{s} .**¹ Signal variations that occur at scale \bar{s} are indicative of *potential* peaks of interest.

When we fix a scale \bar{s} , the resulting $|W_x(\bar{s}, t)|$ is a function of time. The next task is to find the *local maxima* of $|W_x(\bar{s}, t)|$ as t varies. The correspondence between coefficient magnitudes and signal energy proves that, at $s = \bar{s}$, the times when local maxima occur are precisely the times when the energy of scale- \bar{s} variations is concentrated. **Thus peak characterization further requires querying the local maxima at \bar{s} .**

Not all maxima are equally interesting; rather, only those with value above a threshold, since these are indicative of signal variations with large energy concentrated at \bar{s} . **Therefore, the specification only considers those local maxima with A value above a threshold \bar{p} .** Again, \bar{p} is chosen based on knowledge of the signal class.

Maxima in the wavelet spectrogram are not isolated: as shown in [19, Thm. 6.6], when the wavelet ψ is the n^{th} derivative of a Gaussian, the maxima belong to connected curves $s \mapsto \gamma(s)$ that are never interrupted as the scale de-

¹The scale should roughly correspond to the period of a single electrogram in the signal.

creases to 0. These *maxima lines* can be clearly seen in Fig. 5 as being the vertical lines of brighter color. Multiple maxima lines may converge to the same point $(0, t_c)$ in the spectrogram as $s \rightarrow 0$. A celebrated result of Mallat and Hwang [18] shows that *singularities* in the signal always occur at the convergence times t_c . For our purposes, a singularity is a time when the signal undergoes an abrupt change (specifically, the signal is poorly approximated by an $(n + 1)^{\text{th}}$ -degree polynomial at that change-point). **These convergence times are then the peak times that we seek.**

Although theoretically, the maxima lines are connected, in practice, signal discretization and numerical errors will cause some interruptions. Therefore, rather than require that the maxima lines be connected, we only require them to be (ϵ, δ) -connected. Given $\epsilon, \delta > 0$, an (ϵ, δ) -connected curve $\gamma(s)$ is one such that for any s in its domain, $|s - s'| < \epsilon \implies |\gamma(s) - \gamma(s')| < \delta$.

A succinct description of this *Wavelet Peaks with Maxima* (WPM) characterization is then:

- (Characterization WPM) Given positive reals $\bar{s}, \bar{p}, \epsilon, \delta > 0$, a peak is said to occur at time t_0 if there exists a (ϵ, δ) -connected curve $s \mapsto \gamma(s)$ in the (s, t) -plane such that $\gamma(0) = t_0$, $|W_x(s, \gamma(s))|$ is a local maximum along the t -axis for every s in $[0, \bar{s}]$, and $|W_x(\bar{s}, \gamma(\bar{s}))| \geq \bar{p}$.

The choice of values $\bar{s}, \epsilon, \delta$ and \bar{p} depends on prior knowledge of the class of signals we are interested in. Such choices are pervasive and unavoidable in signal processing, as they reflect application domain expertise. Implementations of this characterization can vary in the ways they detect maxima lines $s \mapsto \gamma(s)$, and in their freedom to tighten the degree of connectedness (ϵ, δ) . Such a specification is difficult, if not impossible, to express in temporal logic and time-frequency logic [10]. In contrast, the industry-standard specification language PSL allows arbitrary mixtures of *regular expressions* and temporal logic. So regular expressions seem to be a good starting point. However, they are also not expressive enough for the kind of characterization above. In the next section we show how WPM can be formalized using Quantitative Regular Expressions (QREs).

3.1 Blanking Characterization

For comparison, we modify WPM to obtain an alternative characterization which, instead of considering maxima lines, uses the concept of *blanking*. This characterization, which we call *Wavelet Peaks with Blanking* (WPB), produces a computationally cheaper monitor at the cost of imprecision in peak-detection times. It says that one peak at the most can occur in a time window of size BL samples.

- (Characterization WPB) Given positive reals $\bar{s}, \bar{p} > 0$, a peak is said to occur at time t_0 if $|W_x(\bar{s}, t_0)|$ is a local maximum along t and $|W_x(\bar{s}, t_0)| > \bar{p}$, and there is no peak occurring anywhere in $(t_0, t_0 + BL]$.

Note that since this characterization only considers the one scale \bar{s} , it may be viewed as essentially a time-domain description. The relationship between WPM and WPB can be established as follows. Let t_0 be a peak time found by tracing maxima lines. That is, $(0, t_0)$ is a convergence point for maxima lines. Consider all maxima lines $\{\gamma_i\}$, defined at least on $[0, \bar{s}]$, that converge to $(0, t_0)$ as $s \rightarrow 0$: $\gamma_i(0) = t_0 \forall i$. Let $C > 0$ be a constant s.t. for all i and all s in $[0, \bar{s}]$,

$(s, \gamma_i(s)) \in \{(s, t) \mid |t - t_0| < Cs\}$. The latter is known as *the cone of influence* of $W_x(0, t_0)$. By choosing $BL = 2C\bar{s}$, the two specifications yield the same number of peaks. The difference is that the specification with blanking WPB declares the peak to have occurred at t_0 instead of $t_0 + C\bar{s}$ (the center of the cone). Moreover, when the heart rate accelerates, more true peaks will fall inside the fixed blanking period and will be ignored, leading to under-detection.

4. A QRE PRIMER

Expressing complex quantitative properties of signals with nontrivial time-frequency interaction in logic is challenging, as one usually encounters a lack of expressive power. Our first attempt at specifying Peak Detection (PD) was in logic, and was unsatisfactory; see Section 1.1. We concluded that we needed a language that: 1) Is as close as possible to a formal specification language for bounded-time temporal properties. 2) Allows a rich set of numerical operations. 3) Allows matching of complex patterns in the signal, to select scales and frequencies at which interesting structures exist. 4) Supports the synthesis of time- and memory-efficient implementations.

Since regular expressions (RE) are equivalent to Linear Temporal Logic formulas (LTL) when expressing bounded-time temporal properties, and at least as widely used as LTL, we decided that Quantitative Regular Expressions (QREs) would be a good match for our problem. A Quantitative Regular Expression (QRE) is a symbolic regular expression over a data domain D , augmented with data costs from some cost domain C . A QRE views the signal as a *stream* $w \in D^*$ that comes in one data item at a time. As the RE matches the input stream, the cost (or value) of the QRE is being evaluated.

EXAMPLE 1. Assume that a uniformly-sampled discrete-time voltage signal $y(t)$ comes in as stream of values $(y_i)_{i \in \mathbb{N}}$. In other words, for each i , $y_i = y(iT)$, where T is the sampling period. We want to count the number of samples that exceed a certain threshold $\theta \in \mathbb{R}$. To do this, we need to first use QRE $\text{count} := (y > \theta ? 1) \text{ else } (y \leq \theta ? 0)$ to produce a 1 whenever the current sample exceeds the threshold, and 0 otherwise. This matches the symbolic Regular Expression (RE) $(y > \theta \mid y \leq \theta)$, where the word “symbolic” highlights the fact that the basic match is a predicate on the data item y (here, the predicates are $y > \theta$ and $y \leq \theta$). Second, to count how many samples exceed the threshold, we would like to sum the output stream of this expression; so, we use the quantitative regular expression $\text{aboveTh} := \text{iter-sum}(\text{count})$, which iterates the count QRE and adds each new result to the accumulating sum. In this case, the data domain $D = \mathbb{R}$ and the cost domain $C = \mathbb{N}$. The sum operation combines the costs of subexpressions.

Formally, consider a set of types $\mathcal{T} = \{T_1, T_2, \dots, T_k\}$, a data domain $D \in \mathcal{T}$, a cost domain $C \in \mathcal{T}$, and a parameter set $X = (x_1, x_2, \dots, x_k)$, where each x_i is of type T_i . Then a QRE f is a function

$$\llbracket f \rrbracket: D^* \rightarrow (T_1 \times T_2 \times \dots \times T_k \rightarrow C) \cup \{\perp\}$$

where \perp is the undefined value. Intuitively, if the input string $w \in D^*$ does not match the RE of f , then $\llbracket f \rrbracket(w) = \perp$. Else, $\llbracket f \rrbracket(w)$ is a function from $T_1 \times T_2 \times \dots \times T_k$ to C . When a parameter valuation $\bar{v} \in T_1 \times \dots \times T_k$ is given,

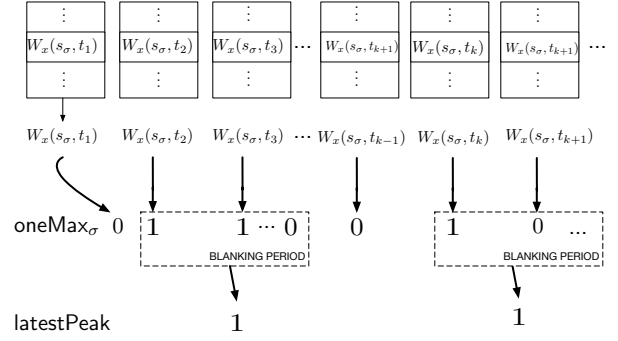


Figure 8: QRE peakWPB

this then further evaluates to a cost value in C , namely $\llbracket f \rrbracket(w)(\bar{v})$. Fig. 6 provides an overview of QREs and their combinators.

QREs can be compiled into efficient *evaluators* that process each data item in time (or memory) polynomial in the size of the QRE and proportional to the maximum time (or memory) needed to perform an *operation* on a set of cost terms, such as addition, least-squares, etc. The operations are selected from a set of operations *defined by the user*. It is important to be aware that the choice of operations constitutes a trade-off between expressiveness (what can be computed) and complexity (more complicated operations cost more). See [2] for restrictions placed on the predicates and the symbolic regular expressions.

The declarative nature of QREs will be important when writing complex algorithms, without having to explicitly maintain state and low-level data flows. But as with any new language, QREs require some care in their usage. Space limitations preclude us from giving the formal definition of QREs. Instead, we will describe what each QRE does in the context of peak detection to give the reader a good idea of their ease of use and capabilities. Fig. 6 illustrates how QREs are defined and what they compute. Readers familiar with QREs will notice that, when writing the QRE expressions, we occasionally sacrifice strict syntactic correctness for the sake of presentation clarity.

5. QRE IMPLEMENTATION OF WPM

A numerical implementation of a CWT returns a discrete set of coefficients. Let $s_1 < s_2 < \dots < s_n$ be the analysis scales and let t_1, t_2, \dots be the signal sampling times. Recall that a QRE views its input as a stream of incoming data items. A data item for WPM is $d = (s_i, t_j, |W_x(s_i, t_j)|) \in D := (\mathbb{R}_+)^3$. We use $d.s$ to refer to the first component of d , and $d.|W_x(s, t)|$ to refers to its last component. The input stream $w \in D^*$ is defined by the values from the spectrogram (see Fig. 5) organized in a column-by-column fashion starting from the highest scale:

$$w = \underbrace{(s_n, t_1, |W_x(s_n, t_1)|), \dots, (s_1, t_1, |W_x(s_1, t_1)|)}_{w_{t_1}} \dots \underbrace{(s_n, t_m, |W_x(s_n, t_m)|), \dots, (s_1, t_m, |W_x(s_1, t_m)|)}_{w_{t_m}}$$

Since the scales $s_i > \bar{s} = s_\sigma$ are not relevant for peak detec-

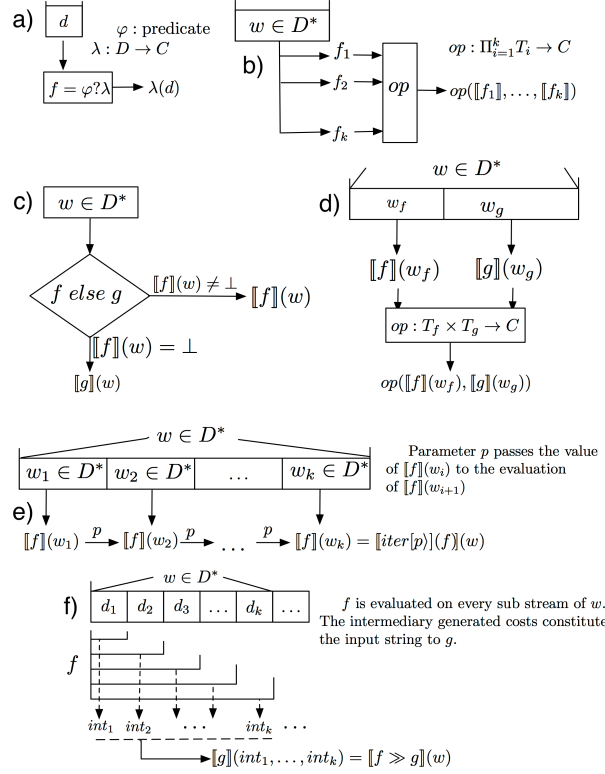


Figure 6: QREs and their combinators. a) Basic QRE $\varphi? \lambda$ matches one data item d and evaluates to $\lambda(d)$ if $\varphi(d)$ is True. b) QRE $op(f_1, \dots, f_k)$ evaluates the k QREs f_1, \dots, f_k on the same stream w and combines their outputs using operation op (e.g., addition). f_i outputs a value of type T_i . c) QRE $f \text{ else } g$ evaluates to f if f matches the input stream; else it evaluates to g . d) QRE $split-op(f, g)$ splits its input stream in two and evaluates f on the prefix and g on the suffix; the two results are then combined using operation op . e) QRE $iter[p](f)$ iteratively applies f on substreams that match it, analogously to the Kleene-* operation for REs. Results are passed between iterations using parameter p . f) QRE $f \gg g$ feeds the output of QRE f into QRE g as f is being computed.

tion (their frequency is too low), we would like to eliminate them from w . Now, for each scale s_i , $i \leq \sigma$, we would like to find the local maxima of $|W_x(s_i, \cdot)|$ which are larger than threshold p_i^2 .

We build the QRE `peakWPM` bottom-up as follows. In what follows, $i = 1, \dots, \sigma$. See Fig. 7.

- QRE `selectCoefi` selects the wavelet coefficient magnitude at scale s_i from the incoming spectrogram column w_t . It must first wait for the entire column to arrive in a streaming fashion, so it matches n data items (recall there are n items in a column – see Fig. 7) and returns as cost $d \cdot |W_x(s_i, t)|$.

$$\text{selectCoef}_i := (d_n d_{n-1} \dots d_1 ? d_i \cdot |W_x(s_i, t)|).$$

- QRE `repeatSelectCoefi` applies `selectCoefi` to the latest column w_t . To do so, it splits its input stream in two: it executes `selectCoefi` on the last column, and ignores all columns that preceded it using $(d^n)^*$. It returns the

selected coefficient $|W_x(s_i, t)|$ from the last column.

$$\text{repeatSelectCoef}_i := \text{split-right}((d^n)^*, \text{selectCoef}_i)$$

Note that `split-right` returns the result of operating on the right-hand side of the split, i.e. the suffix.

- QRE `localMaxi` matches a string of real numbers of length at least 3: $r_1 \dots r_{k-2} r_{k-1} r_k$. It returns the value of r_{k-1} if it is larger than r_k and r_{k-2} , and is above some pre-defined threshold p_i ; otherwise, it returns 0. This will be used to detect local maxima in the spectrogram in a moving-window fashion. In detail:

$$\text{localMax}_i := \text{split-right}(\mathbb{R}^* ? 0, \text{LM}_3) \quad (3)$$

`localMaxi` splits the input string in two: the prefix is matched by \mathbb{R}^* and is ignored. The suffix is matched by QRE `LM3`: `LM3` matches a length-three string and simply returns 1 if the middle value is a local maximum that is above p_i , and returns zero, otherwise.

- QRE `oneMaxi` feeds outputs of QRE `repeatSelectCoefi` to the QRE `localMaxi`.

$$\text{oneMax}_i := \text{repeatSelectCoef}_i \gg \text{localMax}_i$$

² $p_{i=\sigma} = \bar{p}$, $p_{i<\sigma} = 0$, since we threshold only the spectrogram values at scale \bar{s} . After this initial thresholding, tracing of maxima lines returns the peaks.

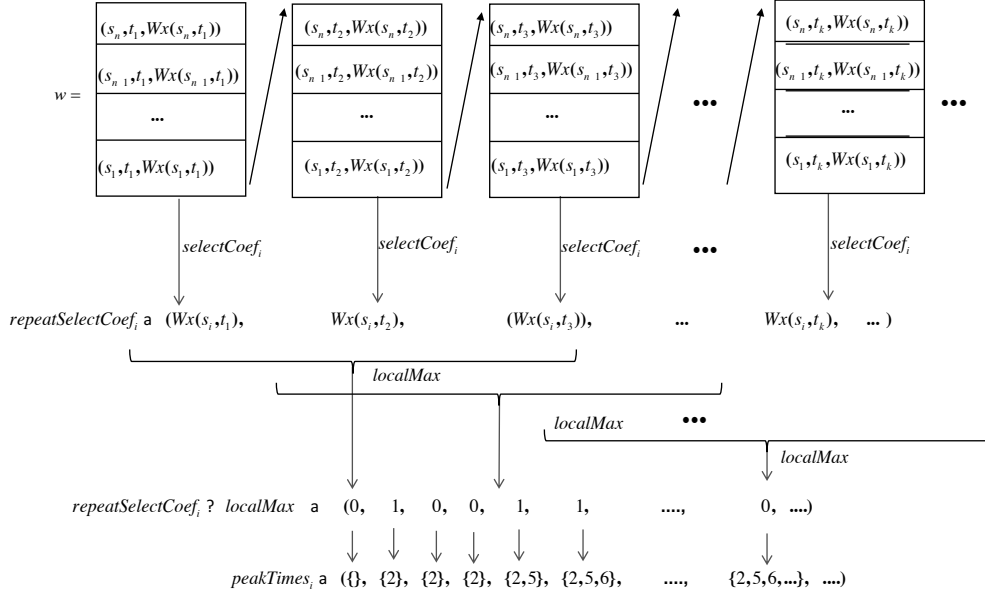


Figure 7: QRE peakWPM

Thus, oneMax_i “sees” a string of coefficient magnitudes $|W_x(s_i, t_1)|, |W_x(s_i, t_2)|, \dots$ generated by (streaming) $\text{repeatSelectCoef}_i$, and produces a 1 at the times of local maxima in this string.

- QRE peakTimes_i collects the times of local maxima at scale s_i into one set.

$$\text{peakTimes}_i := \text{oneMax}_i \gg \text{unionTimes}$$

It does so by passing the string of 1s and 0s produced by oneMax_i to unionTimes . The latter counts the number of 0s separating the 1s and puts that in a set \mathcal{M}_i . Therefore, after k columns w_t have been seen, set \mathcal{M}_i contains all local maxima at scale s_i which are above p_i in those k columns.

- QRE peakWPM is the final QRE. It combines results obtained from scales s_σ down to s_1 :

$$\text{peakWPM} := \text{conn}_\delta(\text{peakTimes}_\sigma, \dots, \text{peakTimes}_1)$$

Operator conn_δ^3 checks if the local maxima times for each scale (produced by peakTimes_i) are within a δ of the maxima at the previous scale.

In summary, the complete QRE is given top-down by:

$$\begin{aligned} \text{peakWPM} &:= \text{conn}_\delta(\text{peakTimes}_\sigma, \dots, \text{peakTimes}_1) \\ \text{peakTimes}_i &:= \text{oneMax}_i \gg \text{unionTimes} \\ \text{oneMax}_i &:= \text{repeatSelectCoef}_i \gg \text{localMax}_i \\ \text{localMax}_i &:= \text{split-right}(\mathbb{R}^*0, \text{LM}_3) \\ \text{repeatSelectCoef}_i &:= \text{split-right}((d^n)^*, \text{selectCoef}_i) \\ \text{selectCoef}_i &:= (d_n \dots d_1 ? d. |W_x(s_i, t)|) \end{aligned}$$

³Operator conn_δ can be defined by recursion as follows:

$$\begin{aligned} \text{conn}_\delta(X, Y) &= \{y \in Y : \exists x \in X : |x - y| \leq \delta\} \\ \text{conn}_\delta(X_k, \dots, X_1) &= \text{conn}_\delta(\text{conn}_\delta(X_k, \dots, X_2), X_1) \end{aligned}$$

5.1 QRE Implementation of WPB

We implement wavelet characterization WPB of Section 3 as QRE peakWPB . See Fig. 8. The input data stream is the same as before.

- QRE oneMax_σ (defined as before) produces a string of 1s and 0s, with the 1s indicating local maxima at scale $\bar{s} = s_\sigma$.
- QRE oneBL matches one blanking duration, starting with the maximum that initiates it. Namely, it matches a maximum (indicated by a 1), followed by a blanking period of length BL samples, followed by any-length string without maxima (indicated by 0*):

$$\text{oneBL} := (1 \cdot (0|1)^{BL} \cdot 0^*)$$

- QRE latestPeak will return a 1 at the time of the latest peak in the input signal.

$$\text{latestPeak} = \text{split-right}(\text{oneBL}^*0, 1?1).$$

It does so by matching all the blanking periods up to this point using oneBL^* and ignoring them. It then matches the maximum (indicated by 1) at the end of the signal.

- QRE peakWPB feeds the string of 1s and 0s produced by oneMax_σ to the QRE latestPeak :

$$\text{peakWPB} = \text{oneMax}_\sigma \gg \text{latestPeak}$$

6. EXPERIMENTAL RESULTS

In this section, we show the results of running the above QREs on real patient data, obtained from a dataset of intra-cardiac electrograms. We also specified a peak detector available in a commercial ICD [23] as QRE peakMDT , and show the results for comparison purposes. Our objective is two-fold: highlight the challenges involved in peak detection, an essential signal-processing task in many medical

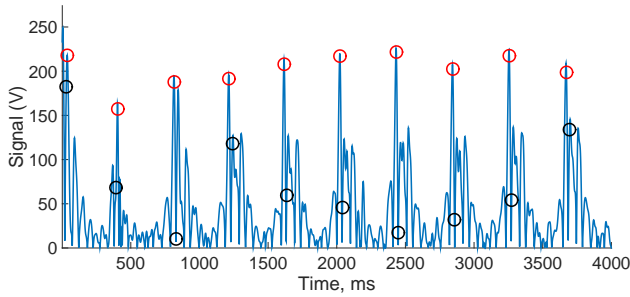


Figure 9: Rectified EGM of a VT showing peakWPM-detected peaks as red circles and peakWPB as black circles ($\bar{p} = 400$)

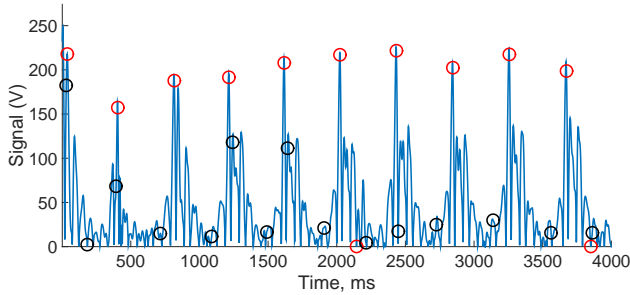


Figure 10: peakWPM (red), peakWPM (black) on VT signal ($\bar{p} = 150$)

devices; and to demonstrate the benefit in adopting a more sophisticated, time-frequency characterization of peaks such as **peakWPM**, over simpler time-domain ones like **peakWPB** and **peakMDT**. This, in turn, prompts the use of more expressive formalisms, like QREs, for analysis.

6.1 Comparing peakWPM and peakWPB

Figs. 9–10 present one rectified EGM signal of a Ventricular Tachycardia (VT) recorded from a patient. Circles (indicating detected time of peak) show the result of running **peakWPM** (red circles) and **peakWPB** (black circles). These results were obtained for $\bar{s} = 80$, $BL = 150$, and different values of \bar{p} . The first setting of \bar{p} (Fig. 9) for both QREs was chosen to yield the best performance. This is akin to the way cardiologists set the parameters of commercial peak detectors: they observe the signal, then set the parameters. We refer to this as the *nominal setting*. Ground-truth is obtained by having a cardiologist examine the signal and annotate the true peaks.

We first observe that the peaks detected by **peakWPM** match the ground-truth; i.e., the nominal performance of **peakWPM** yields perfect detection. This is not the case with **peakWPB**. Next, one can notice that the time precision of detected peaks with **peakWPM** is higher than with **peakWPB** due to maxima lines tracing down to the zero scale. *Note also that the results of peakWPM are stable for various parameters settings.* Improper thresholds \bar{p} or projection scales \bar{s} degrade the results only slightly (compare locations of red circles on Fig. 9 with Fig. 10). By contrast, **peakWPB** detects additional false peaks (compare black cir-

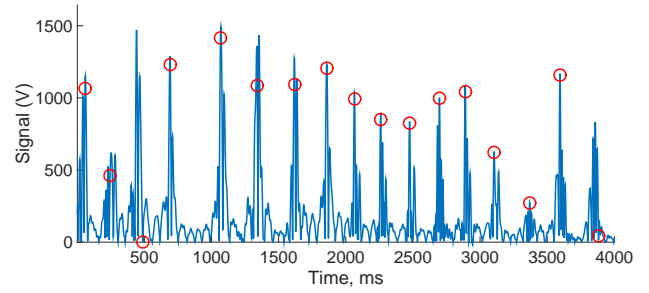


Figure 11: peakWPM with nominal settings ($\bar{s} = 100$, $\bar{p} = 300$) for VT with atrial flutter

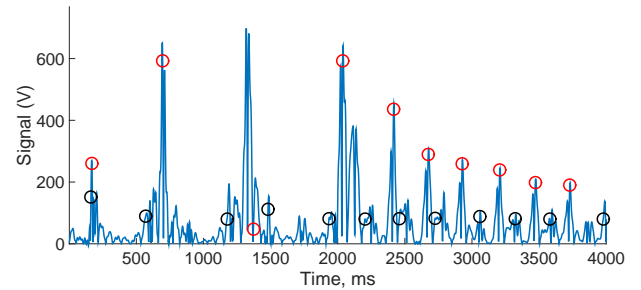


Figure 12: Results of running peakWPM (red) and peakMDT (black) with nominal settings on the VF with NSR signal.

cles in Figs. 9 and 10). Finally, note that peak times found with **peakWPB** precede the actual peak times (e.g., black circles in Fig. 10). This can be explained by the fact that **peakWPB** declares a peak when the local maximum of coefficient values passes the threshold value for the first time. This point does not always correspond to an actual peak.

Red circles in Figs. 11 and 12 show the results of running **peakWPM** on signals with even more variability in signal amplitude and frequency. Fig. 11 displays *Ventricular Fibrillation*, a disorganized state of cardiac electrical activity. Fig. 12 shows VT with atrial flutter. Most peaks were detected correctly (even low-energy peaks around 250ms and 3400ms in Fig. 11). Note, however, that peaks around 500ms and 1400ms were detected with reduced precision.

6.2 Results of commercial peakMDT

The QRE **peakMDT** works almost perfectly with nominal parameters settings on any Normal Sinus Rhythm (NSR) signal (see Fig. 13). NSR is the “normal” heart rhythm. The detected peak times are slightly early because **peakMDT** declares a peak when the signal exceeds a time-varying threshold, rather than when it reaches its maximum. Using the same nominal parameters on more disorganized EGM signals with higher variability in amplitude, such as VF, does not produce proper results; see the black circles in Fig. 12. Other possible values of parameters (blanking period and minimal threshold) improve **peakMDT** results, but still produce false positives; see Fig. 14 (additional peaks are detected around 1500ms and 2250ms). *Analysis of this and other similar results leads to the conclusion that it is impossible to set up the peakMDT parameters in order to get 100%*

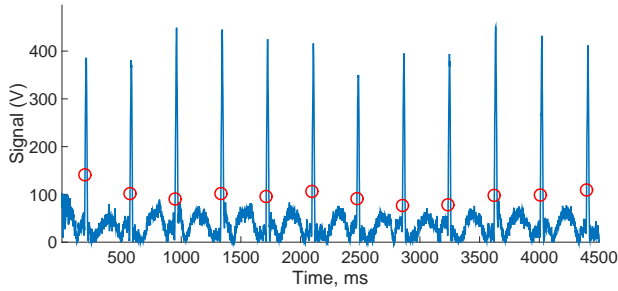


Figure 13: Results of peakMDT with nominal settings running on an NSR signal

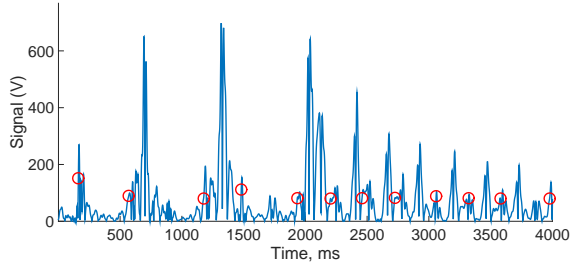


Figure 14: Results of peakMDT with non-nominal settings ($BL = 250\text{ms}$, $\text{minThr} = 80$) running on the VF with NSR signal

precision in the results. In contrast, it was shown that for *peakWPM*, such a setting is possible, and it uses the nominal parameter values (see Fig. 12, red circles).

7. RELATED WORK

Temporal logic [21] and its extensions [17, 10] have gained much success as concise formal specification languages for expressing complex temporal behavior. One popular extension is Signal Temporal Logic (STL) [17] which is designed for the specification of temporal, real-time properties over real-valued signals. STL has been used in many applications ranging from monitoring and analysis of mixed-signal integrated circuits [13] to the discrimination of medical signals [6, 3]. One important limitation of STL is the lack of operators to specify time-frequency properties. These are necessary for peak detection and discrimination [22]. In [5], STL was augmented with a signal value *freeze operator* that allows one to refer to certain time-points in the signal when a formula φ is true (e.g., local minima). Although this approach can be used to express oscillatory patterns such as damped oscillations, it is not possible to use it to discriminate oscillations within a particular frequency range. Furthermore, the associated monitoring algorithm is too complex to be implemented in practice (especially in hardware), since it involves expensive operations over polytopes.

The spectrogram of a signal can be represented as a 2D map (from time and scale to amplitude) and one may think to employ a spatial-temporal logic such as SpaTeL [12] or Signal Spatio-Temporal Logic (SSTL) [20] on spectrograms. However, both of their underlying spatial models, graph structures for SSTL and quadrees for SpaTeL, are not appropriate for this purpose.

Logics for describing frequency and temporal properties have been proposed, including Time-Frequency Logic (TFL) in [10] and the approach in [8]. TFL combines temporal logic operators in the time domain with predicates in the frequency domain. TFL is not sufficiently expressive for peak detection because it lacks the necessary mechanisms to quantify over variables or to freeze their values. Any extension of TFL in this direction would necessarily increase the computational complexity of the generated monitoring algorithm. For this reason, in this paper, we follow a generic, transducer-oriented approach [9, 4, 7] by building upon QREs which enjoy efficient monitoring properties.

8. CONCLUSIONS AND FUTURE WORK

We have presented a general wavelet-based characterization of *peaks* in cardiac electrograms, along with two peak-detection algorithms (*peakWPM* and *peakWPB*) based on this characterization. Both algorithms, and a commercial time-domain one from Medtronics, are readily expressible in QREs. An accuracy and sensitivity study using real patient data shows that the wavelet-based *peakWPM* algorithm outperforms the other two algorithms, yielding results that are on par with those provided by a cardiologist.

Like linear temporal logic (LTL), Quantitative Regular Expressions (QREs) view a signal as a linear data stream (coming in one data item at a time). QREs, however, have additional properties that make them very well-suited for the quantitative analysis of discrete-time real-domain signals in general, and medical-device signals in particular. 1) They can incorporate an arbitrary set of user-defined operations, which gives them an expressive power not matched by the quantitative semantics of temporal logics. 2) They have compilation algorithms that produce efficient implementations, namely, both memory and runtime are polynomial in the size of the QRE and proportional to the cost of the most expensive (user-defined) operation [2]. In ongoing work, we use the peak characterization presented here in to specify various arrhythmia discriminators of ICDs in QREs.

Just like an RE has an equivalent machine model (DFA), a QRE has an equivalent machine model in terms of transducers. This transducer-based characterization of QREs points to various types of analysis of a QRE's correctness and efficiency beyond testing. As future work, we intend to investigate three of them in the context of medical devices.

Probabilistic analysis. First, assume a probabilistic model of the QRE's input: input strings w are drawn from some probability distribution supported on D^* . For medical devices, such a model can be learned from data. We may then perform a statistical analysis of the output of the QRE under such an input model to estimate the average number of computations required to obtain an answer, the probability of computing a given value, etc. Such statistics have direct relevance to the study of medical devices. For an ICD, the first statistic is predicting how long it takes the ICD to detect a fatal arrhythmia. The second is computing the probability of an incorrect detection by the ICD.

Distance between QREs. Here one needs an appropriate notion of distance between two signals (e.g., EGMs or the output of *peakWPM*, *peakWPB* or *peakMDT*), and gradually generalize this to a distance between two QREs. For example, a generalized Skorokhod distance was used in [1] to quantify the value and time mismatches between trajectories of hybrid systems. In [15], a Skorokhod distance was used to

quantify these mismatches between continuous signals by allowing the application of time distortions to minimize their pointwise distance. The work in [16] extends the definition of this distance to sets of signals in the form of *flow-pipes*. Recent work in [14] defines a proper distance between two discrete-time, discrete-domain signals, and between a signal of this type and a Signal Temporal Logic (STL) specification. Given a set of EGMs whose peaks have been annotated by a team of cardiologists (this set of ground-truth is always going to be finite), we can readily compare the *outputs* of two QREs processing these same inputs (for example, **peakWPM** vs **peakWPB** vs **peakMDT**) using the notion of distance in [14]. Generalizing this comparison to two QREs will be more challenging, and it should both generalize the finitary case and take advantage of the structure of the QREs.

Energy calculations. We may compute or approximate the energy consumption of an algorithm that is expressed as a QRE. This is natural and straightforward in the QRE formalism, since the energy consumption can be viewed as just another quantity computed by the QRE. Alternatively, we may label the transitions of the equivalent transducer by “energy terms”, and treat the transducer as a weighted automaton⁴. We can then leverage analysis techniques of weighted automata to analyze the energy consumption. Energy considerations are crucial to implanted medical devices that must rely on a battery, and which require surgery to replace a depleted battery.

9. REFERENCES

- [1] H. Abbas and G. Fainekos. Formal property verification in a conformance testing framework. In *Proc. of MEMOCODE 2014*. ACM, 2014.
- [2] R. Alur, D. Fisman, and Raghathan. Regular programming for quantitative properties of data streams. In *Proc. of ESOP 2016: the 25th European Symposium on Programming Languages and Systems*, volume 9632 of *LNCS*, pages 15–40. Springer, 2016.
- [3] E. Bartocci, L. Bortolussi, and G. Sanguinetti. Data-driven statistical learning of temporal logic properties. In *Proc. of FORMATS 2014*, volume 8711 of *LNCS*, pages 23–37. Springer, 2014.
- [4] L. Bozzelli and C. Sánchez. Foundations of boolean stream runtime verification. In *Proc. of RV 2014: the 5th International Conference on Runtime Verification*, volume 8734 of *LNCS*, pages 64–79, 2014.
- [5] L. Brim, P. Dluhos, D. Safránek, and T. Vejpustek. STL-*: Extending signal temporal logic with signal-value freezing operator. *Inf. Comput.*, 236:52–67, 2014.
- [6] S. Bufo, E. Bartocci, G. Sanguinetti, M. Borelli, U. Lucangelo, and L. Bortolussi. Temporal logic based monitoring of assisted ventilation in intensive care patients. In *Proc. of ISO LA*, volume 8803 of *Lecture Notes in Computer Science*, pages 391–403. Springer, 2014.
- [7] P. Caspi, D. Pilaud, N. Halbwachs, and J. Plaice. Lustre: A declarative language for programming synchronous systems. In *Proc. of POPL 1987: the Fourteenth Annual ACM Symposium on Principles of Programming Languages*, pages 178–188. ACM Press, 1987.
- [8] A. Chakarov, S. Sankaranarayanan, and G. Fainekos. Combining time and frequency domain specifications for periodic signals. In *Runtime Verification: Second International Conference, San Francisco, CA, USA*, pages 294–309, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.
- [9] B. D’Angelo, S. Sankaranarayanan, C. Sánchez, W. Robinson, B. Finkbeiner, H. B. Sipma, S. Mehrotra, and Z. Manna. LOLA: Runtime monitoring of synchronous systems. In *Proceedings of the 12th International Symposium of Temporal Representation and Reasoning (TIME 2005)*, pages 166–174. IEEE Computer Society Press, 2005.
- [10] A. Donzé, O. Maler, E. Bartocci, D. Nickovic, R. Grosu, and S. A. Smolka. On Temporal Logic and Signal Processing. In *Proc. of ATVA 2012, Thiruvananthapuram, India, October 3-6*, volume 7561 of *LNCS*, pages 92–106, 2012.
- [11] G. E. Fainekos and G. J. Pappas. Robustness of temporal logic specifications for continuous-time signals. *Theor. Comput. Sci.*, 410(42):4262–4291, 2009.
- [12] I. Haghighi, A. Jones, Z. Kong, E. Bartocci, R. Grosu, and C. Belta. Spatel: A novel spatial-temporal logic and its applications to networked systems. In *Proc. of HSCC ’15: the 18th International Conference on Hybrid Systems: Computation and Control*, pages 189–198. ACM, 2015.
- [13] S. Jaksic, E. Bartocci, R. Grosu, R. Klobhofer, T. Nguyen, and D. Nickovic. From signal temporal logic to FPGA monitors. In *MEMOCODE 2015, Austin, TX, USA, September 21-23, 2015*, pages 218–227, 2015.
- [14] S. Jakić, E. Bartocci, R. Grosu, and D. Ničković. Quantitative monitoring of STL with edit distance. In Y. Falcone and C. Sánchez, editors, *Proc. of RV 2016: the 16th International Conference on Runtime Verification*, volume 10012 of *LNCS*, pages 201–218. Springer, 2016.
- [15] R. Majumdar and V. S. Prabhu. Computing the Skorokhod distance between polygonal traces. In *Proc. of HSCC 2015: the 18th International Conference on Hybrid Systems*, pages 199–208. ACM, 2015.
- [16] R. Majumdar and V. S. Prabhu. Computing distances between reach flowpipes. In *Proc. of HSCC 2016: the 19th International Conference on Hybrid Systems*, pages 267–276. ACM, 2016.
- [17] O. Maler and D. Nickovic. Monitoring temporal properties of continuous signals. In *Proc. of FORMATS/FTRTFT 2004*, volume 3253 of *LNCS*, pages 152–166, 2004.
- [18] S. Mallat and W. L. Hwang. Singularity detection and processing with wavelets. *IEEE Transactions on Information Theory*, 38(2):617–643, March 1992.
- [19] S. G. Mallat. *A Wavelet Tour of Signal Processing, Third Edition: The Sparse Way*. Academic Press, 2008.
- [20] L. Nenzi, L. Bortolussi, V. Ciancia, M. Loreti, and M. Mieke. Qualitative and quantitative monitoring of spatio-temporal properties. In *Proc. of RV 2015: the*

⁴In more detail, if we only consider the DFA that underlies the transducer (this is the DFA that accepts the regular expression of the QRE), and augment its transitions by energy costs, then we have a weighted automaton.

- 6th International Conference on Runtime Verification*, volume 9333 of *LNCS*, pages 21–37. Springer, 2015.
- [21] A. Pnueli. The temporal logic of programs. *Foundations of Computer Science, IEEE Annual Symposium on*, 0:46–57, 1977.
- [22] F. Scholkmann, J. B. Boss, and M. Wolf. An efficient algorithm for automatic peak detection in noisy periodic and quasi-periodic signals. *Algorithms*, 5(4):588–603, 2012.
- [23] R. X. Stroobandt, S. S. Barold, and A. F. Sinnaeve. *Implantable Cardioverter - Defibrillators Step by Step*. Wiley, 2009.
- [24] C. D. Swerdlow, S. J. Asirvatham, K. A. Ellenbogen, and P. A. Friedman. Troubleshooting implanted cardioverter defibrillator sensing problems I. *Circulation: Arrhythmia and Electrophysiology*, 7(6):1237–1261, 2014.